Final Project Report: DOE Award FG02-04ER25606

Overlay Transit Networking for Scalable, High Performance Data Communication across Heterogeneous Infrastructure

1. Introduction: An Infrastructure Problem for Data Intensive, Asynchronous Collaboration

As the flood of data associated with leading edge computational science continues to escalate, the challenge of supporting the distributed collaborations that are now characteristic of it becomes increasingly daunting. The chief obstacles to progress on this front lie less in the synchronous elements of collaboration, which have been reasonably well addressed by new global high performance networks, than in the asynchronous elements, where appropriate shared storage infrastructure seems to be lacking. The recent report from the Department of Energy on the emerging "data management challenge" [1] captures the multidimensional nature of this problem succinctly:

> "Data inevitably needs to be buffered, for periods ranging from seconds to weeks, in order to be controlled as it moves through the distributed and collaborative research process. To meet the diverse and changing set of application needs that different research communities have, large amounts of non-archival storage are required for transitory buffering, and it needs to be widely dispersed, easily available, and configured to maximize flexibility of use. In today's grid fabric, however, massive storage is mostly concentrated in data centers, available only to those with user accounts and membership in the appropriate virtual organizations, allocated as if its usage were non-transitory, and encapsulated behind legacy interfaces that inhibit the flexibility of use and scheduling. This situation severely restricts the ability of application communities to access and schedule usable storage where and when they need to in order to make their workflow more productive." (p.69f)

One possible strategy to deal with this problem lies in creating a storage infrastructure that can be universally shared because it provides only the most generic of asynchronous services. Different user communities then define higher level services as necessary to meet their needs. One model of such a service is a Storage Network, analogous to those used within computation centers, but designed to operate on a global scale. Building on a basic storage service that is as primitive as possible, such a Global Storage Network would define a framework within which higher level services can be created. If this framework enabled a variety of more specialized middleware and supported a wide array of applications, then interoperability and collaboration could occur based on that common framework.

The research in *Logistical Networking* (*LN*) carried out under the DOE's SciDAC program tested the value of this approach within the context of several SciDAC application communities. Below we briefly describe the basic design of the LN storage network and some of the results that the Logistical Networking community has achieved.

2. Infrastructure and Fundamental Services

The core services of any Global Storage Network are the allocation of persistent buffers and transfer of data between such buffers. Using these operations, a wide range of operations can be implemented, including

- Creation of striped, replicated files encoded for erasure or error correction.

- Application-specific prestaging and caching of communication channels
- Flexible movement of data across multiple media and through multiple network paths.

These basic services are building blocks in the creation of tools which enable and support a variety of application communities, but not necessarily in a uniform way. Logistical Networking, with the *Internet Backplane Protocol (IBP)* as its fundamental service [2, 3], is a framework which we propose as the basis of a Global Storage Network. IBP implements a generic, best effort network storage service that can scale globally. Two key characteristics of IBP storage are:

- Allocations of IBP storage are limited in size and duration. An IBP allocation request can be refused in response to over-allocation and the storage resource can be revoked when the lease expires. Also, an IBP server may be restricted to use only idle disk resources ("soft" storage), ensuring that the host machine does not over commit resources.
- Semantics of IBP storage are weaker than the typical storage service. IBP storage resource can be transiently unavailable or even permanently lost. With "soft" storage allocation semantics, resource can be revoked at any time before expiration.

IBP storage is managed by servers called "depots", on which clients perform remote storage operations. The depot was designed for simplicity and robustness by using a stateless protocol. IBP clients view a depot's storage resources as a collection of byte arrays. Clients initially obtain the use of a byte array by making a storage allocation on a depot. If the allocation is successful (depending on size and duration requested as well as the storage resources available), the depot returns three cryptographically secure URLs, called capabilities, to the client: one for reading, one for writing, and one for management. Capabilities may be passed from client to client, requiring no registration from or notification to the depot. The synchronous (blocking) IBP client calls fall into three different groups as shown in Table 1.

Table 1. Synchronous IBP Client API

Depot Management	`IBP_status`
Storage Management	`IBP_manage`
Data Transfer	`IBP_store IBP_load IBP_(m)copy`

Because of the limitations on allocation size, duration and semantics, IBP does not directly implement strong storage services such as conventional files. Like IP's "best effort" datagram service, IBP is designed to be the generic common service at the "waist of the hourglass" on which all stronger storage-based services must build and by means of which all storage-based applications can achieve interoperability (Figure 1). Basic middleware tools for using this common network storage service have already been developed and are available at http://loci.cs.utk.edu. For example, the XML encoded exNode was created to aggregate and manage primitive IBP byte arrays. The Logistical Runtime System (LoRS) consists of a set of tools and associated APIs that build on exNodes in order to enable users to implement files and other storage abstractions that can draw on a pool of depots and can possess a wide range of characteristics, such as large size (via fragmentation), fast access (via caching), and reliability (via replication). LoRS tools also implement some transport layer services such as checksums, encryption, compression, and erasure codes, at the end-points.

2.1. The Logistical Distribution Network (LoDN)

When a file is uploaded into the Logistical Network, the file content is broken into smaller data blocks. Each data block is then replicated and the replicas are distributed across multiple IBP depots. Distributing fragmented replicas of file content over multiple storage locations helps to ensure that stored content will remain accessible in case of a machine or network outage. Currently the distribution of data blocks is determined according to geographical parameters. Each registered IBP depot provides its geographical position (latitude and longitude) to the L-Bone, a directory and resource discovery service catalog of publicly accessible IBP storage depots. When storing data, clients may query the L-Bone for depots with specific characteristics, including minimum storage capacity, duration policy, proximity, etc. When a user uploads a file into the Logistical Network, an exNode is generated. The exNode is an XML encoded metadata file that accommodates the aggregation of individual storage allocations into a "network file" by holding the metadata (where each data block is stored, when its storage "lease" expires, etc.) necessary to manage distributed content. The exNode aggregates individual IBP allocations, thereby allowing the use of multiple allocations to achieve strong characteristics not offered by any single allocation (large file size, extended storage duration, reliability and fault tolerance, etc.)

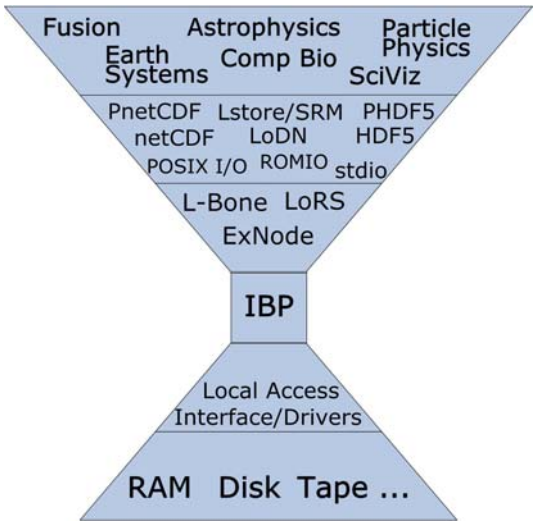

Figure 1: The Logistical Networking storage stack. IBP provides interoperability for applications and storage technologies.

When a file is uploaded with the Logistical Distribution Network (LoDN), the resulting exNode is stored on the LoDN server. An active service operates on all exNodes in the LoDN server to maintain the integrity of the corresponding stored data. LoDN gives fast access to stored data. At download, individual data blocks from a fragmented, distributed file can be retrieved from different depots, with different blocks being downloaded in parallel. LoDN also provides fast throughput by utilizing the proximity of depots to upload and download sites. At upload, you can choose the geographical location where your file is stored; you may store the data in a depot within your neighborhood or place the data on depots close to remote download sites. The LoDN download client automatically retrieves data from the closest depots storing replicas of the desired content.

3. Middleware

The usefulness of LN depends on layers of middleware that implement valuable services over IBP. Such tools create structure by managing capabilities using exNodes and managing depots to enable resource discovery. On top of this layer, standard file I/O tools and interfaces have been ported to take advantage of the generality of LN.

Implementations of standard APIs using LN mechanisms are important for at least two reasons. First and foremost, they allow legacy applications to be converted from working on traditional file systems to working with LN resources without the need to modify the application code. The other reason is that they provide a point of control for implementing optimizations and other adaptations of the services and capabilities that LN provides. Below we briefly characterize some key APIs that have already been converted.

3.1. NetCDF, HDF5, stdio, POSIX I/O

Based on Unidata's NetCDF, LoCI Lab has developed NetCDF/L which adds LN capabilities. Like the traditional NetCDF, NetCDF/L can store data on a local filesystem, but it can also store data on the global logistical network. By specifying a local filename or a LoRS URL (i.e. lors://), the user controls

where the data is stored. NetCDF/L is compatible with CDF version 2 (CDF2) which supports files up to 8 exabytes.

The libxio library, developed in collaboration with the DiDaS project in the Czech Republic, provides a standard Unix I/O interface (i.e. open(), close(), read(), write(), etc.) to access local files as well as logistical technology-based "network files" (exNodes). Researchers can port their Unix I/O applications to add LN capabilities by adding 12 lines of code and recompiling. Like LoRS, libxio supports 64-bit file offsets, user configurable multi-threading, and more. In current work, the stdio library and NCSA's HDF5 are also being ported to run over LoRS.

3.2. ROMIO

Jonghyun Lee of Argonne National Lab is developing an experimental version of the ROMIO Abstract Device IO layer (ADIO on LN) that will implement MPI-IO using the LoRS library. This will allow MPI programs to read and write files described by exNodes, and to interoperate with other tools and programs that use the LoRS and exNodes libraries. This will, for instance, provide interoperability between parallel NetCDF and HDF5 that have been built on top of MPI-IO and the sequential versions of these libraries that we have ported directly as described above.

3.3. Storage Resource Manager for the Open Science Grid

A team of developers at the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University, led by Paul Sheldon and Alan Tackett, is collaborating with LoCI Lab to develop disk- and tape-based data management tools based on LN that are compatible with the Storage Resource Manager (SRM) interface [4]. The SRM interface was defined through a community process led and organized by the Scientific Data Management Center. The ACCRE development team includes Kevin McCord, Surya Pathak. The target for their tools are the Open Science Grid (OSG), and in particular High Energy Physics data distribution problems.

4. Applications

4.1. Astrophysics: Terascale Supernova Initiative

The Terascale Supernova Initiative (TSI) has been running simulations on the large supercomputers at Oak Ridge National Laboratory (RAM and Phoenix) and generating datasets which are growing to multiple terabytes in size. The data must be moved to the primary postprocessing and visualization site as well as stored to overcome faults. However, the scale of the data will not allow long term archival storage for all of it – after a period of analysis and visualization it will have to be discarded in order to make room for the output of successive runs. Up to this point the mode of use of LN by TSI has been as a means to obtain high speed multithreaded data transport and limited-duration data storage. Data files have simply been uploaded from the supercomputer where they are generated to LN storage resources and then downloaded onto the cluster at North Carolina State University where they are then analyzed and visualized. However, with the size of datasets increasing to the point that the duration of download is prohibitive, TSI researchers are evaluating the use of NetCDF and HDF libraries that can access portions of the data directly through LN without downloading the entire dataset to the local file system.

4.2. Fusion Energy: Gyrokinetic Particle Simulation of Turbulent Transport in Burning Plasma

Scott Klasky of Princeton Plasma Physics Laboratory and Viraj Bhat and Manish Parashar of Rutgers University's Computer Science Dept. have been working in collaboration with LoCI Lab on the development of new techniques for buffering and transferring data generated by simulations running on large supercomputers at NERSC (Seaborg) and ONRL (Phoenix) to PPPL for analysis and visualization [5]. This work takes advantage of the presence of interoperable IBP depots at PPPL and at NERSC and ORNL, allowing data to be transferred directly from the compute nodes to the PPPL depots up to the speed permitted by the institutional firewall (currently limited to 100Mbps). Once

that link is saturated (or in case of depot or network failure) the remaining data flow (which is generated at a total rate of over 400Mbps) is written to depots local to the generating computing center (NERSC or ORNL respectively). Only if all depots are unavailable is the data written to the local file system. Thus, after a data generating run much of the data is actually located at PPPL, much has been written to depots at the computing center, and generally a small amount is located on the local file system of the supercomputer.

4.3. Scientific Visualization

Scientific visualization is a crosscutting application area that is critical to many SciDAC projects. The goal of creating an infrastructure capable of supporting remote viewing and analysis of simulation data by a team of scientists spread at different locations is an important part of SciDAC's long term vision. Jian Huang of UTK and Jinzhu Gao of ORNL have been working with LoCI lab to make use of LN technology in realizing this goal. Previous research led by Prof. Huang showed that it is feasible to use an IBP-based storage network for the distributed sharing of data in remote visualization across the wide area Internet. Because of application control over buffering, the latency as perceived by a user is comparable to accessing the entire data set across a local area network. [6]. More recent work integrates data buffering in the network with visualization operations performed on the buffered data *in situ* on the network storage nodes, using a large number of network computing units in parallel while still obtain scalable speedups. Achieving this result with LN will create a distributed system for visualization that is highly deployable and easily usable by a large community of users

5. Conclusion

Logistical Networking represents an innovative approach to the problem of provisioning community infrastructure for data intensive asynchronous collaboration. The fundamental infrastructure, network services and middleware have been defined and promulgated; additional layers of middleware have now been added to enable application development and porting. Substantial application communities, within and outside SciDAC, are now considering or have already adopted LN as a strategy for managing persistent distributed state.

[1] Mount R. *The office of science data management challenge: Report from the doe office of science data management workshops*. In: Department of Energy; 2005.

[2] Beck M, Moore T, Plank JS. *An end-to-end approach to globally scalable network storage*. In: Proceedings of SIGCOMM 2002; August 19-23; Pittsburgh, PA; 2002. 339-346.

[3] Bassi A, Beck M, Moore T, Plank JS, Swany M, Wolski R, et al. *The internet backplane protocol: A study in resource sharing*. FGCS 2003;19(4):551-562.

[4] Shoshani A, Sim A, Gu J. *Storage resource managers: Middleware components for grid storage*. In: Nineteenth IEEE Symposium on Mass Storage Systems (MSS '02); 2002.

[5] Bhat V, Klasky S, Atchley S, Beck M, McCune D, Parashar M. *High performance threaded data streaming for large scale simulations*. In: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (Grid 2004); Pittsburgh, PA USA; 2004.

[6] Ding J, Huang J, Beck M, Liu S, Moore T, Soltesz S. *Remote visualization by browsing image based databases with logistical networking*. In: Proceedings of SC2003; Phoenix, AZ; 2003.